Center for Power Electronics Systems
A National Science Foundation Engineering Research Center
Virginia Tech, University of Wisconsin - Madison, Rensselaer Polytechnic Institute
North Carolina A&T State University, University of Puerto Rico - Mayaguez

PEBB Plug and Play Hardware and Software Control Architecture

Dushan Boroyevich, Jerry Francis, Fred Wang, Stephen Edwards

Presentation at

AEPS Review

Warrenton, VA, November 5-7, 2002

Presentation Outline

- **System Architecture** — Jerry Francis
  - What is PnP
  - What does the PnP architecture look like?
- Control Hardware
  - Digital hardware to support control functions
- Power Hardware
  - Demonstration testbed
- Control Software
  - PnP modular software architecture

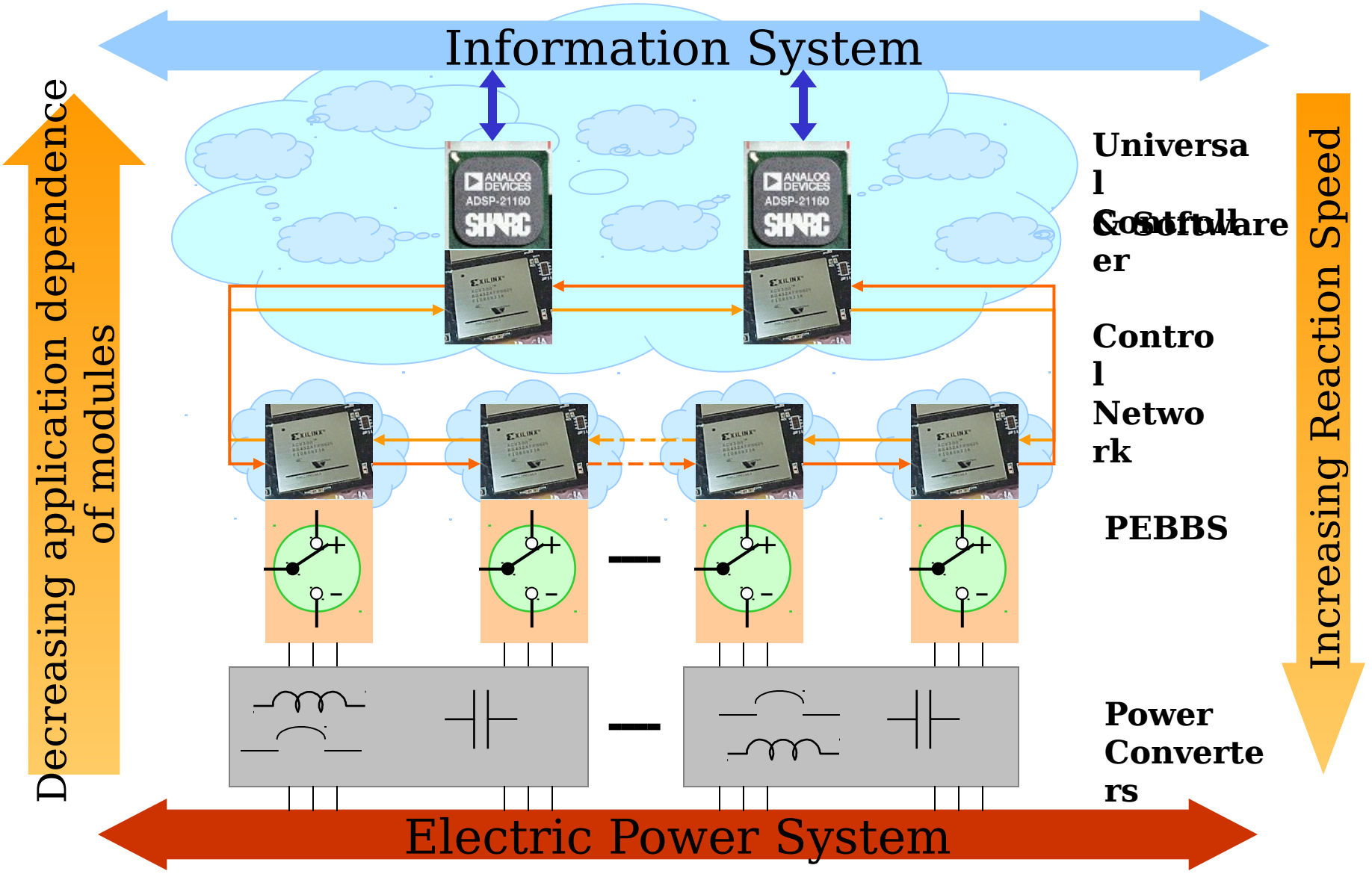
Benefits of PnP (1)

- Plug and Play (PnP) system architecture
  - **Flexible:** Develop COTS modules that can be combined to control applications without customizing each module
    - Modules are interchangable
    - Modules are not application specific
  - **Reconfigurable:** System configures modules depending on application.
  - **Fast:** Meet the requirements of control and information bandwidth

Benefits of PnP (2)

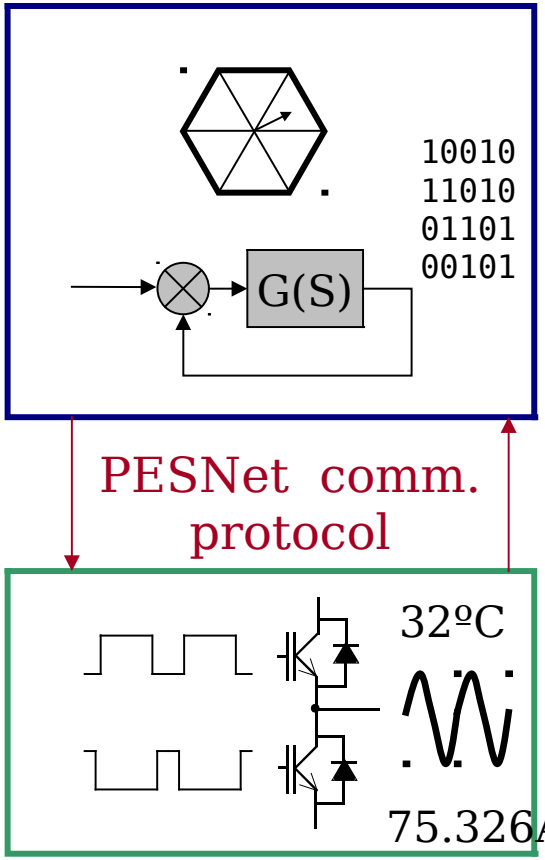
- Plug and Play (PnP) system architecture
  - **Control Hierarchy:** Control is separated into several levels depending on temporal, functional, and spatial distribution
  - **Common Design Approach:** Modules are now generic and no longer specific to application
    - Development effort is more focused on application than developing control hardware

PnP System Architecture



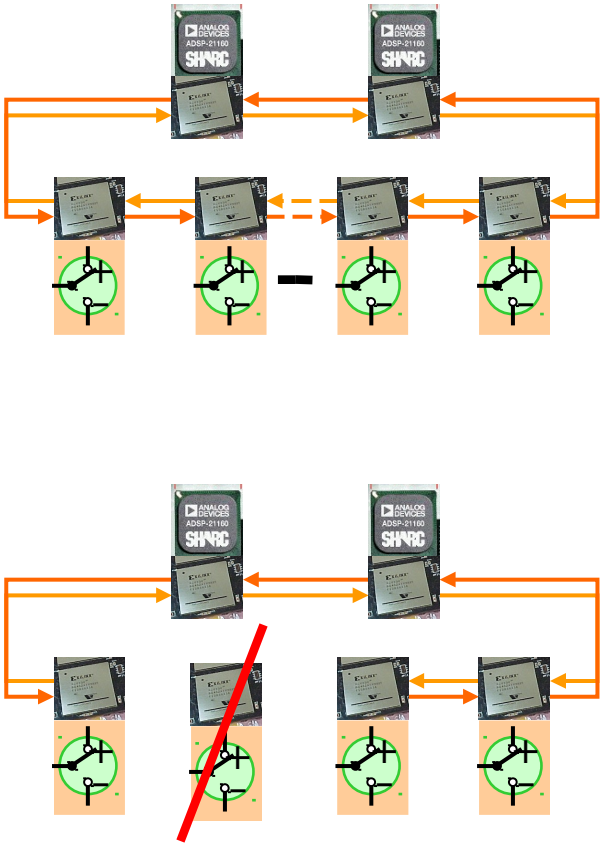
Control Hardware

- Application Manager
  - Universal Controller
  - Application independent
  - Digital system
- Hardware Manager
  - Hardware specific
  - Analog and Mixed-signal
  - Gate Drivers, Sensors
  - Local Control Functions



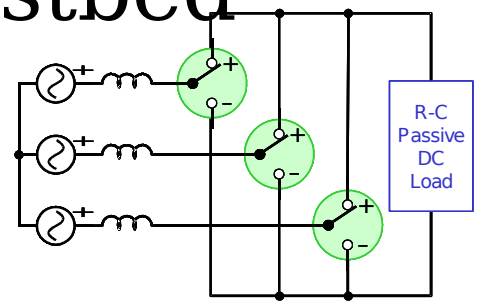
Communication Protocol

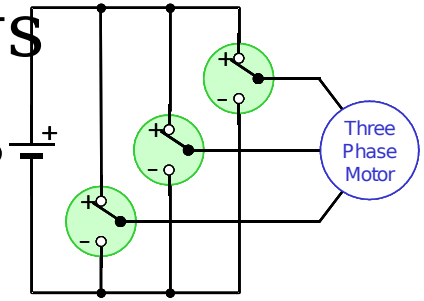
- Fiber Optic Network
- Fault tolerant
  - Dual ring, reroutable
- Robust
  - Variable number of nodes
  - Variable packet size
- Tight Synchronization
  - Synchronizes switching action

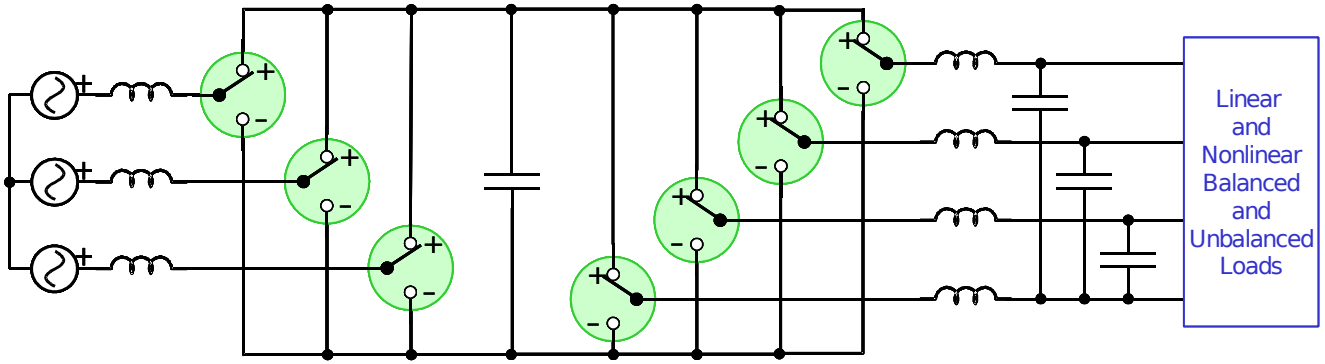


Power Stage

- 100 kW Demonstration Testbed
- Reconfigurable
- Eight Phases
  - 4 Hard-Switching Phase Legs
  - 4 Soft-Switching Phase Legs
- Multiple Topologies

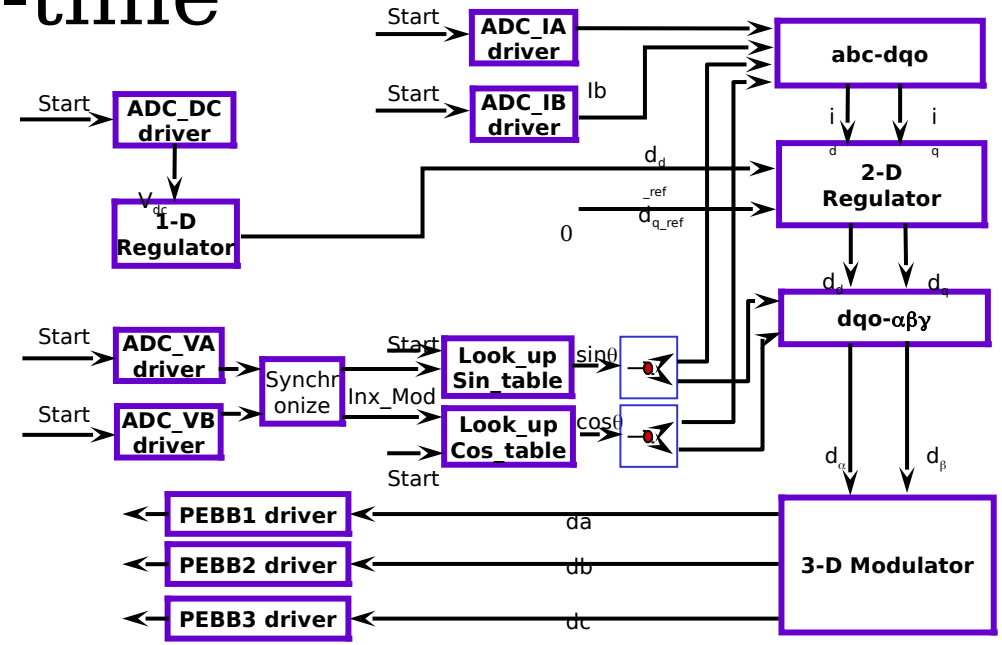






Control Software

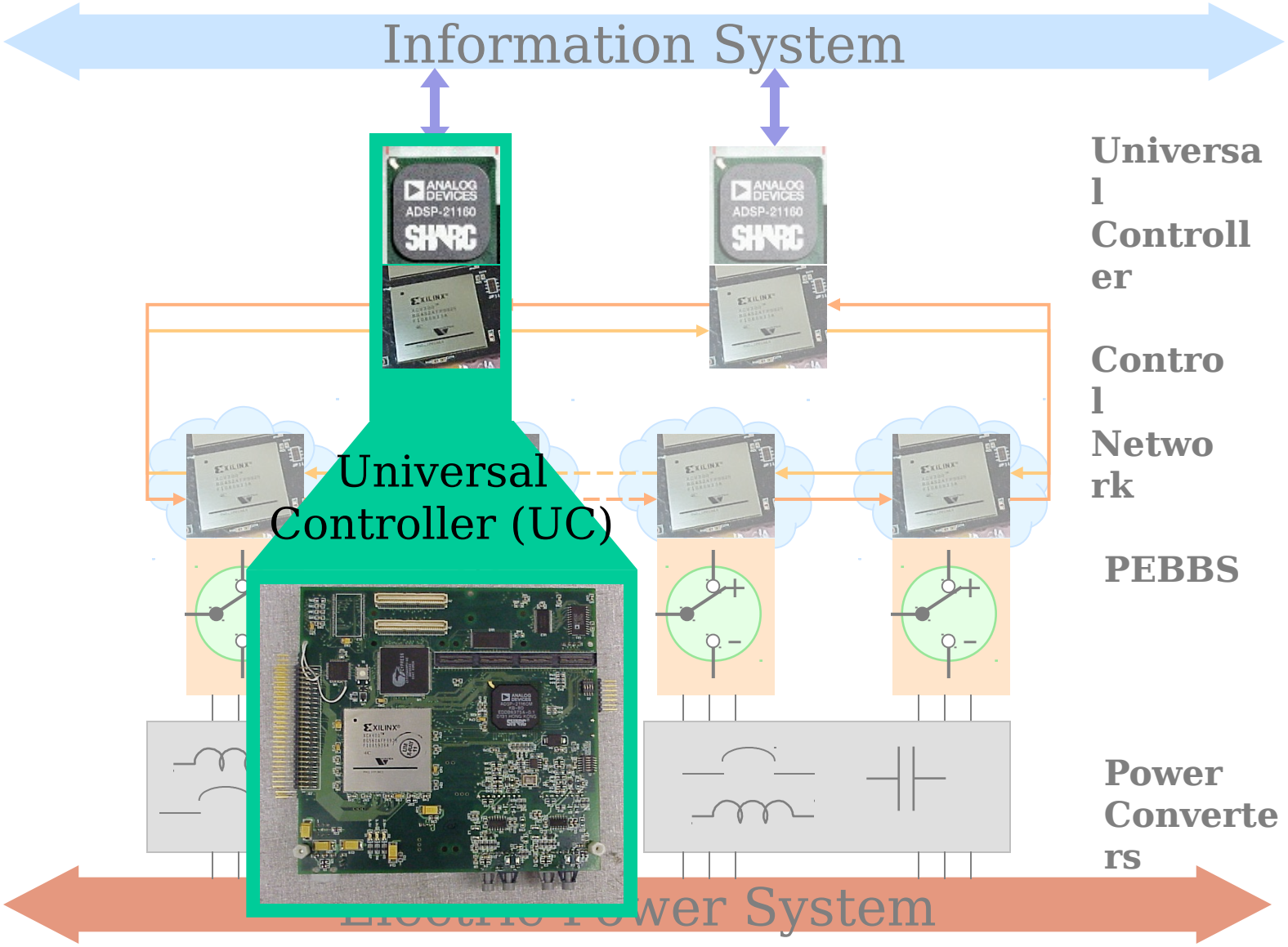
- Modular, reconfigurable, and flexible dataflow building-blocks
- Elementary Control Objects
- Custom real-time OS (DARK)



Presentation Outline

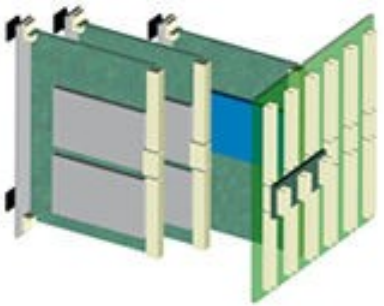
- **System Architecture**
  - What is PnP
  - What does the PnP architecture look like?
- **Control Hardware** — Jerry Francis
  - Digital hardware to support control functions
- Power Hardware
  - Demonstration testbed
- Control Software
  - PnP modular software architecture

Universal Controller

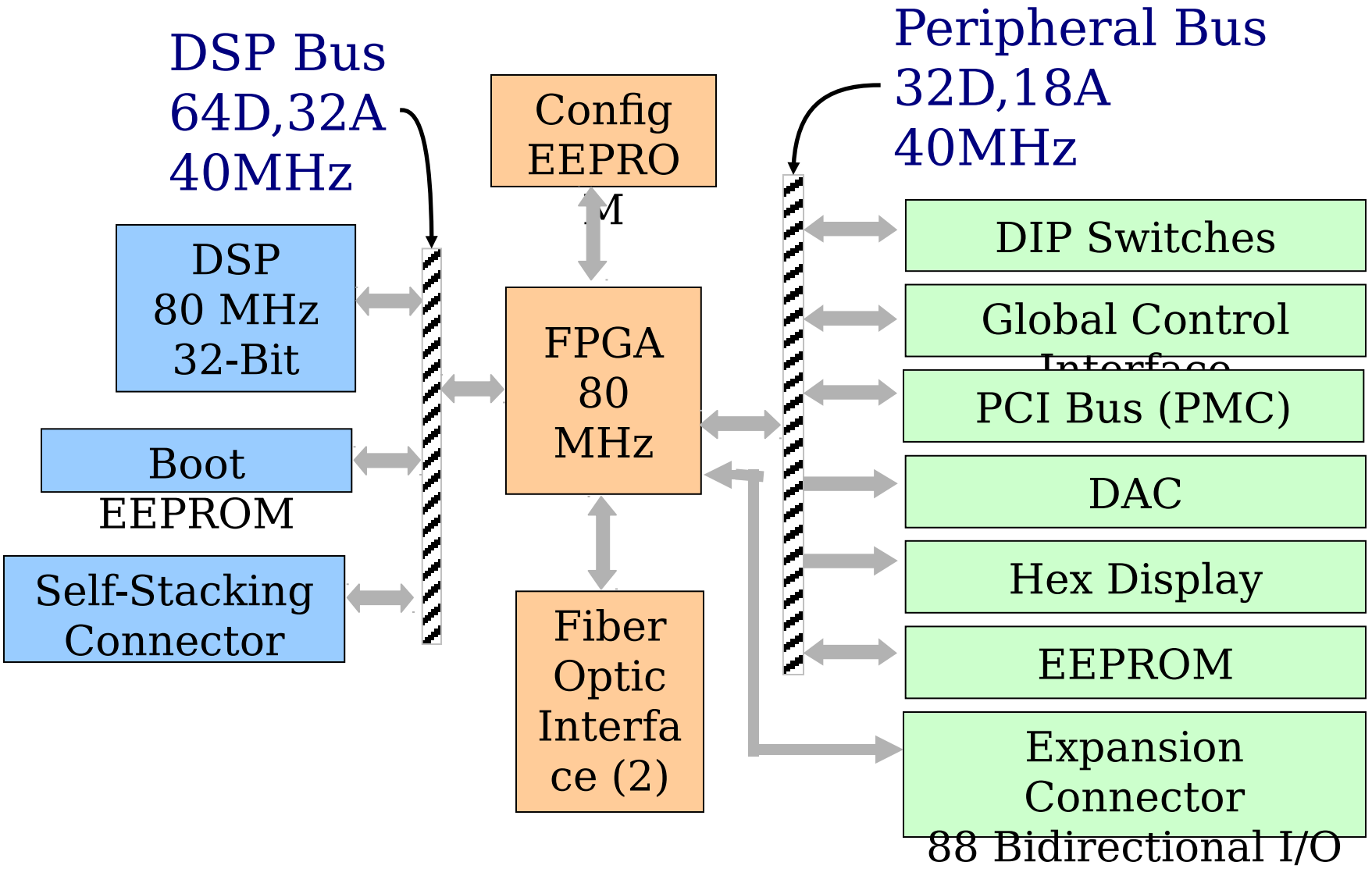


UC Design Objectives

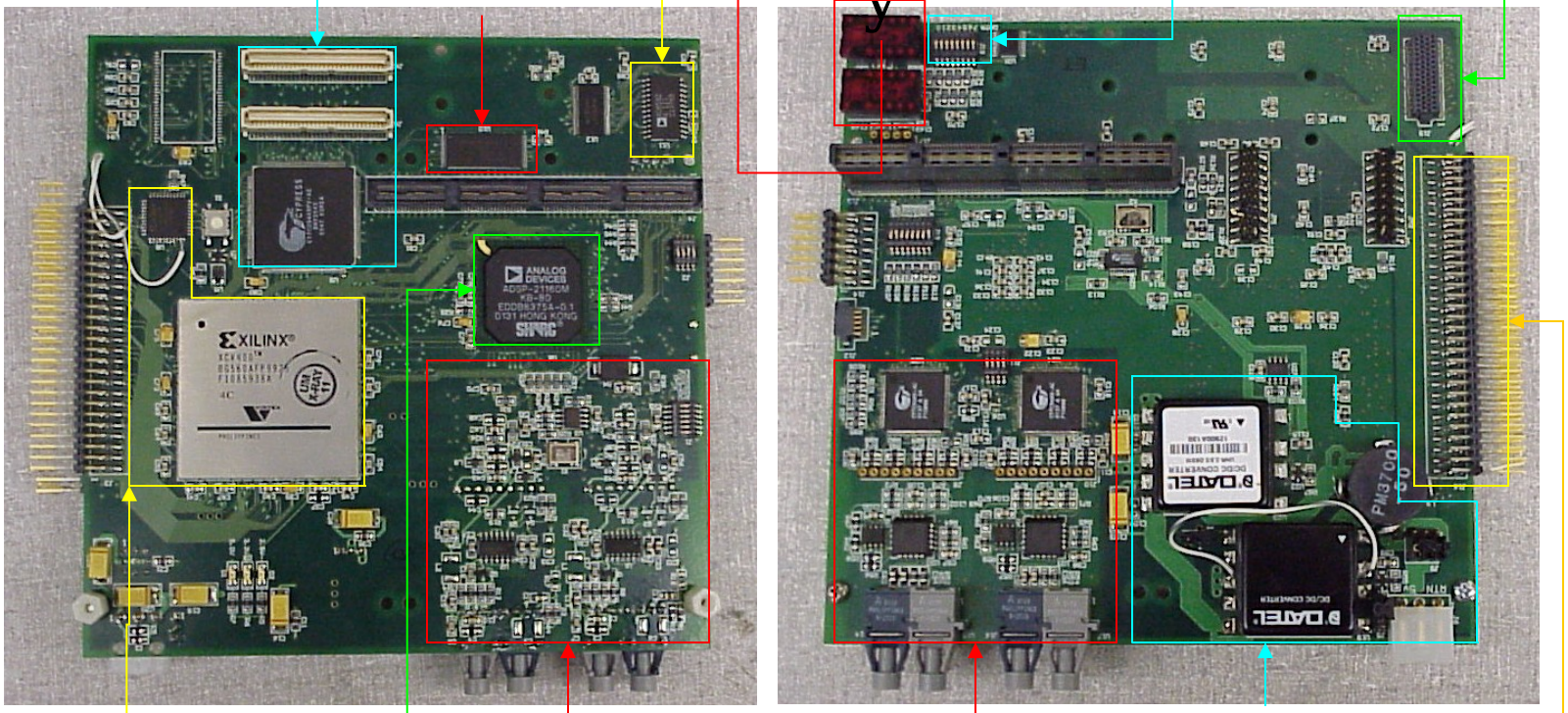
- Flexible
  - Large FPGA and DSP for advanced programs
- Scalable
  - Self-stacking connector
- Compatible
  - Global Controller Interface
  - PESNet Optical Network
  - PCI (PMC)
  - Expansion interfaces



UC Block Diagram



UC Features



STATUS: UC Built and Tested. Second version in production.

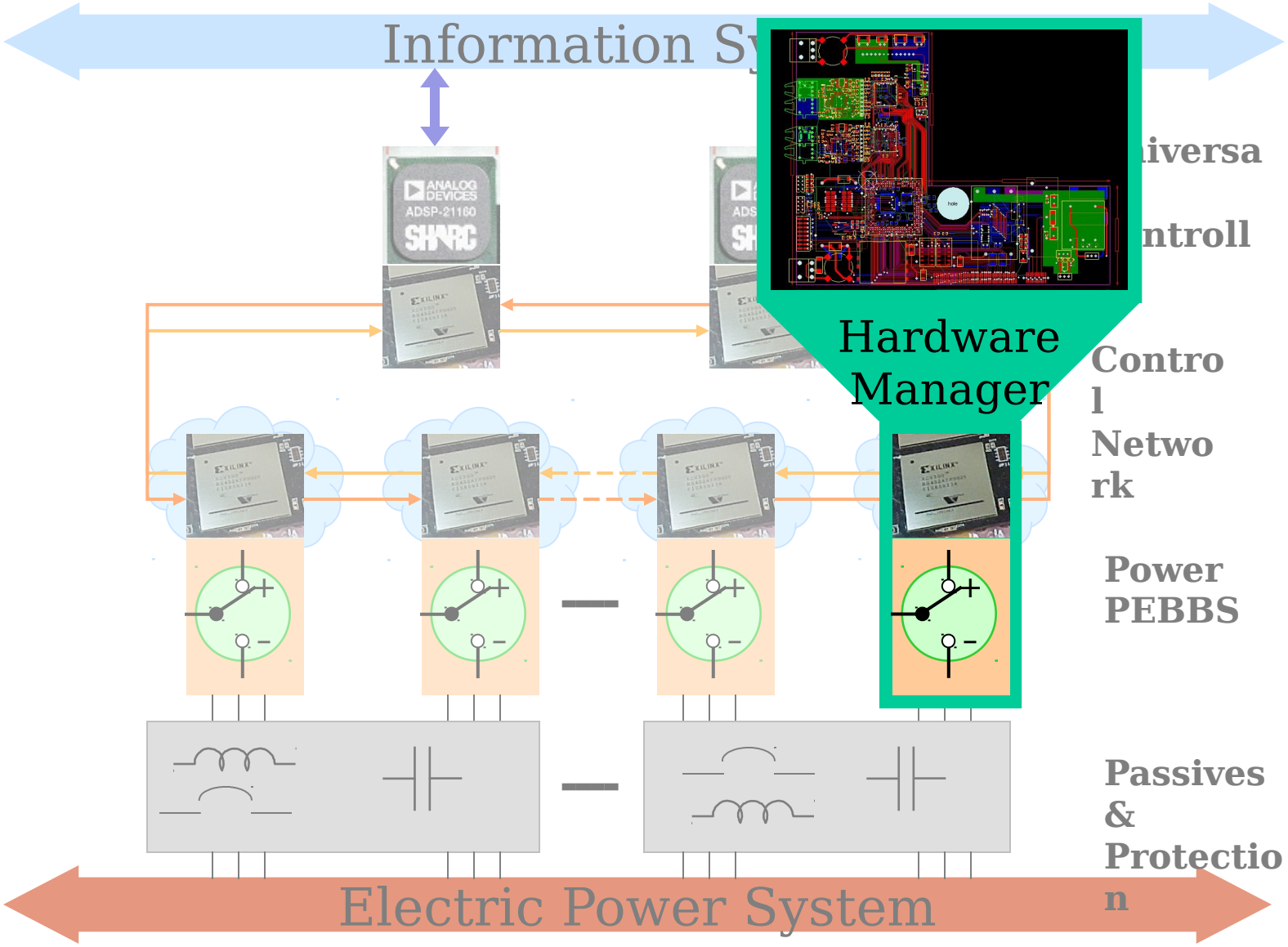
PESNet v. 2.2

- 125 MBps
- Fault tolerant fiber optic ring
- Adjustable packet size
- Multi-master System
- Forward scheduling algorithm
- Fault tolerant network clock
- Topology independent

STATUS: Protocol designed.

Need to finish implementation and verify functionality.

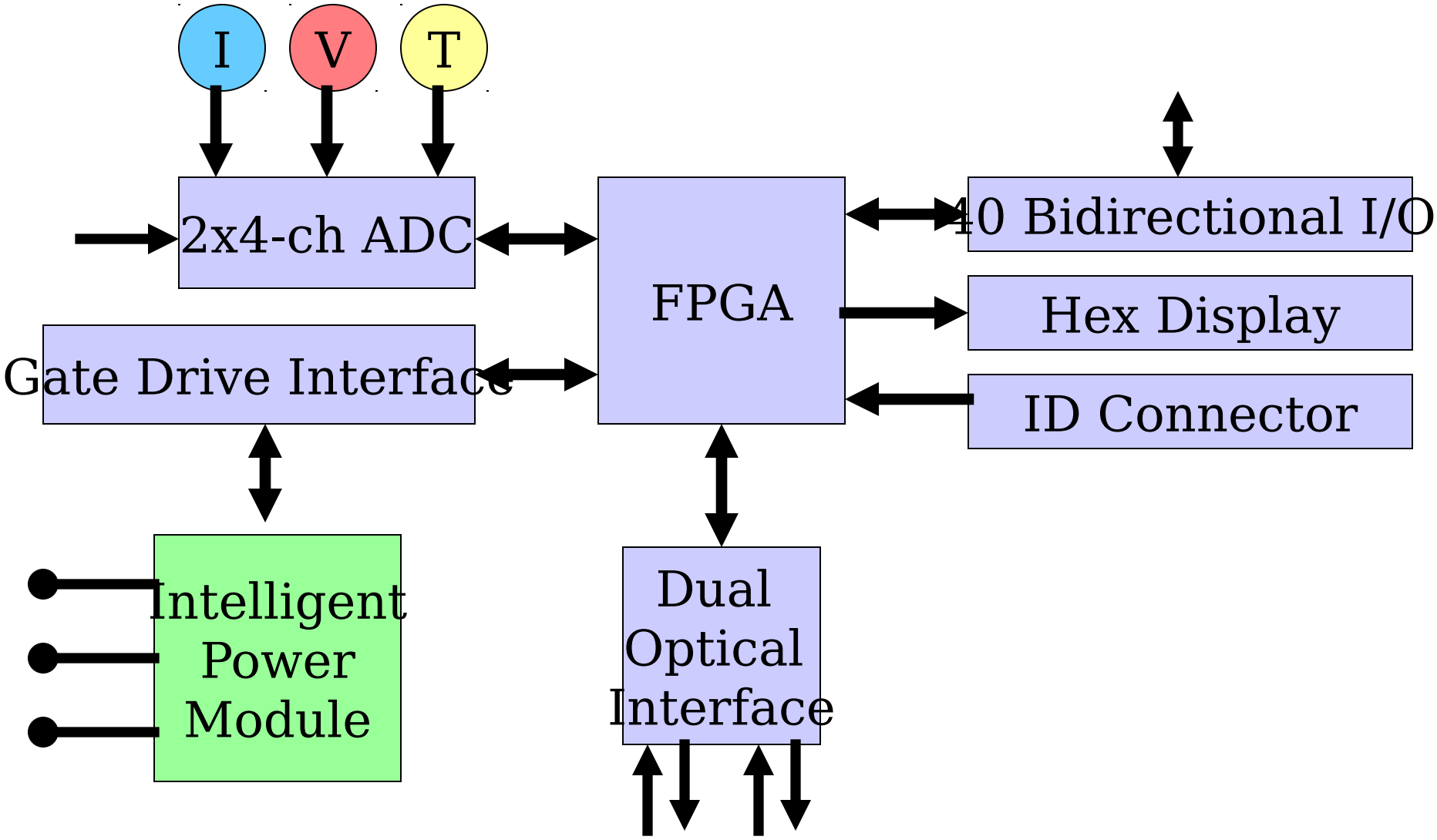
Hardware Manager



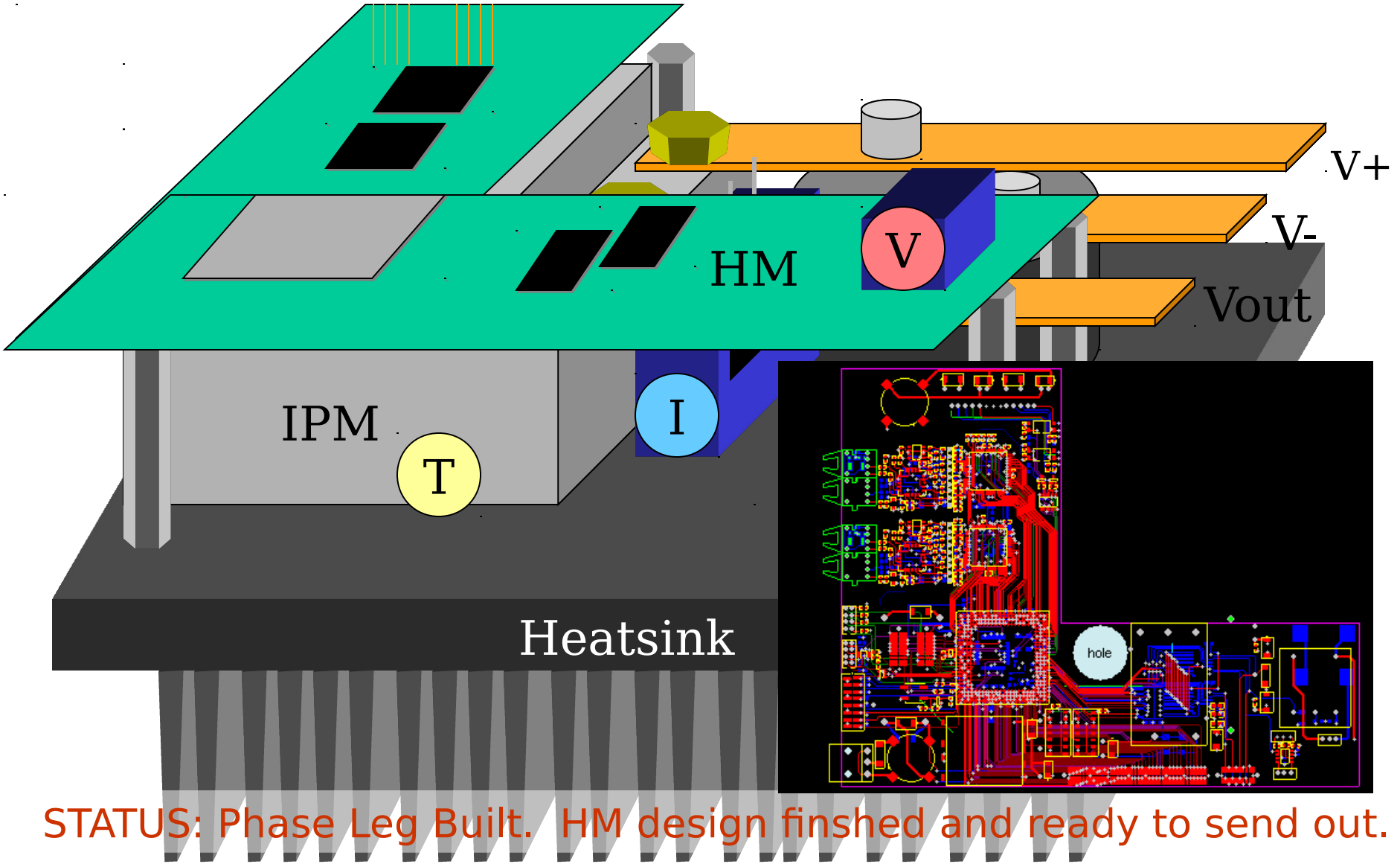
CPES HM Design Objectives

- Single Phase Control Aspects
  - Modulation
  - Protection
  - Soft-Switching Control
  - Current control
- Visual indication of status
- Data acquisition
- Connectivity
- In-Circuit Testability
- In-Circuit Programmability (PnP)

HM Block Diagram



PEBB Physical Design

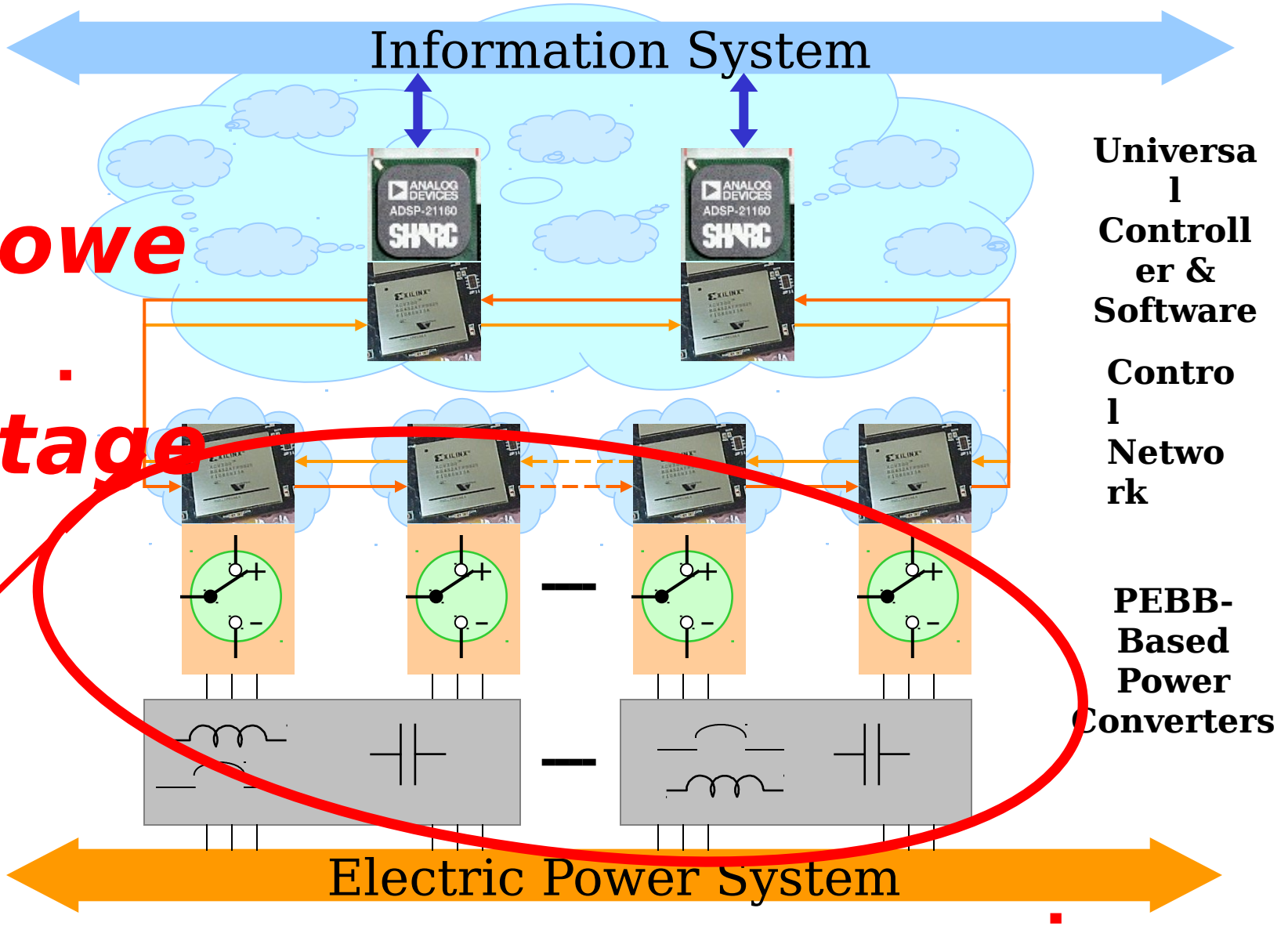


STATUS: Phase Leg Built. HM design finshed and ready to send out.

Presentation Outline

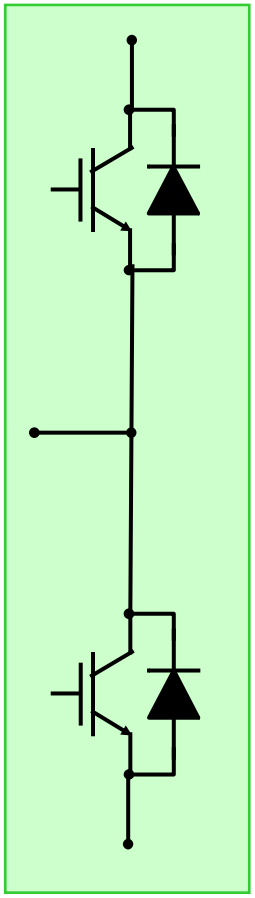
- **System Architecture**
  - What is PnP
  - What does the PnP architecture look like?
- Control Hardware
  - Digital hardware to support control functions
- **Power Hardware** —Fred Wang
  - Demonstration testbed
- Control Software
  - PnP modular software architecture

PnP System Architecture



Power Electronics Building Block

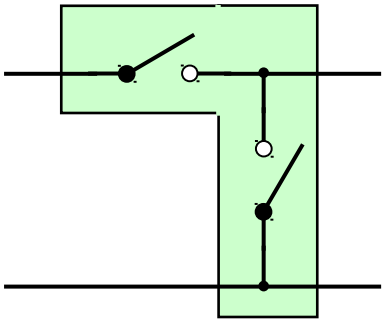
- Common power stage component(s) found in different converter topologies
- Appropriate combination of PEBB's enables the realization of multiple converter topologies
- A feasible PEBB for voltage-source topologies is one converter phase-leg or converter pole
- Study applies to other PEBB topologies



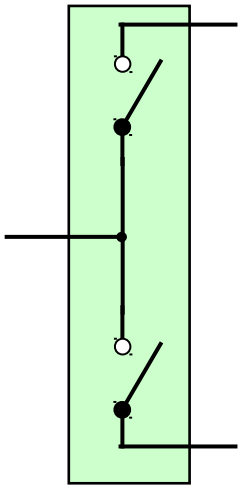
IGBT Voltage-Source PEBB

Voltage-Source Topologies

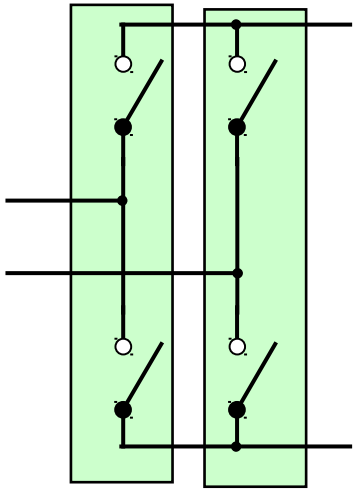
Single-Ended

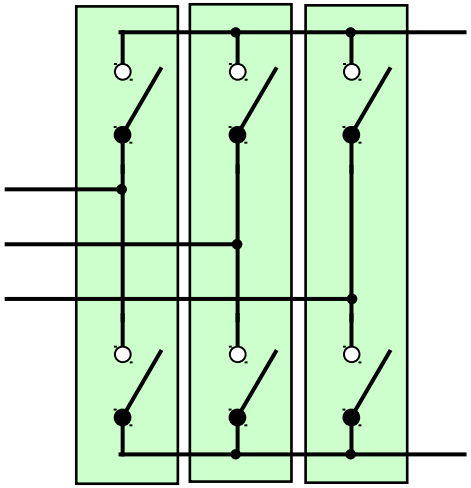


Half-Bridge

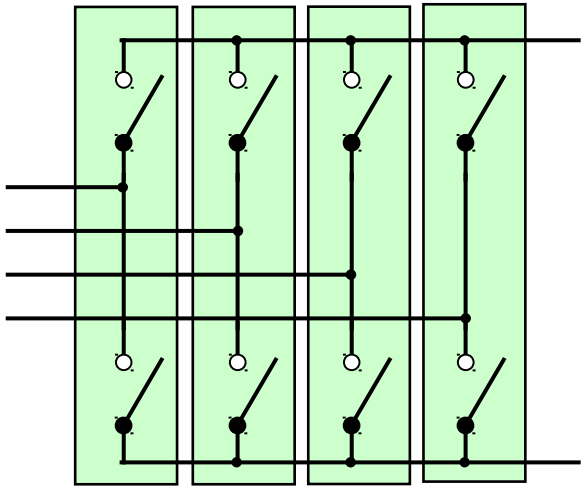


Full-Bridge

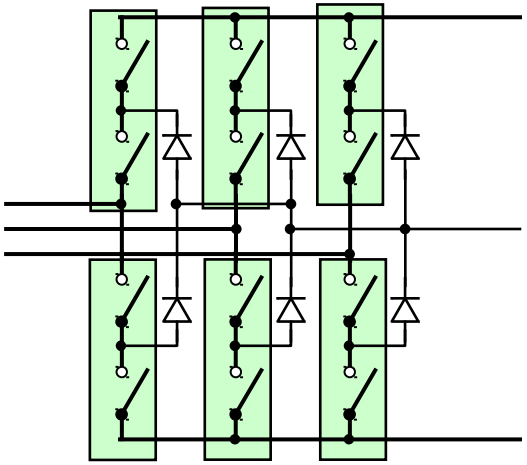




Three-Phase



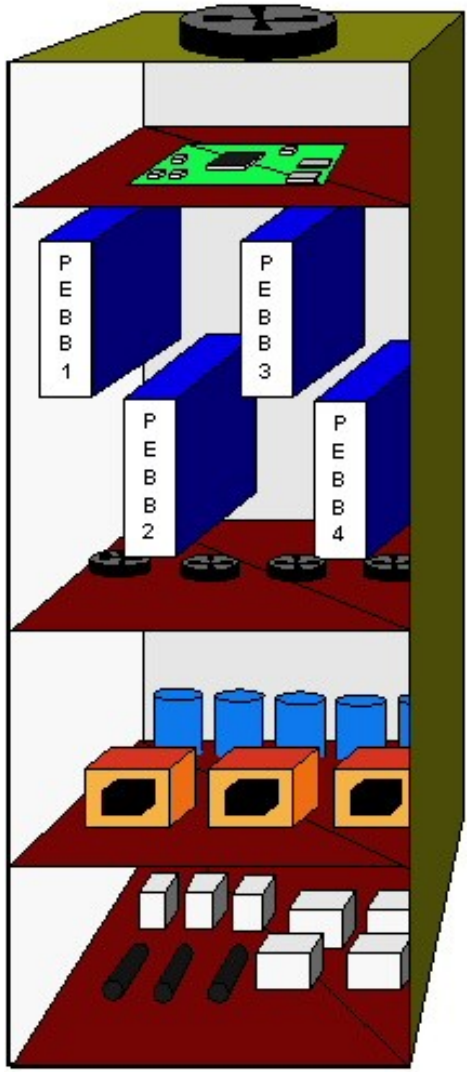
Multi-Phase



Multi-Level

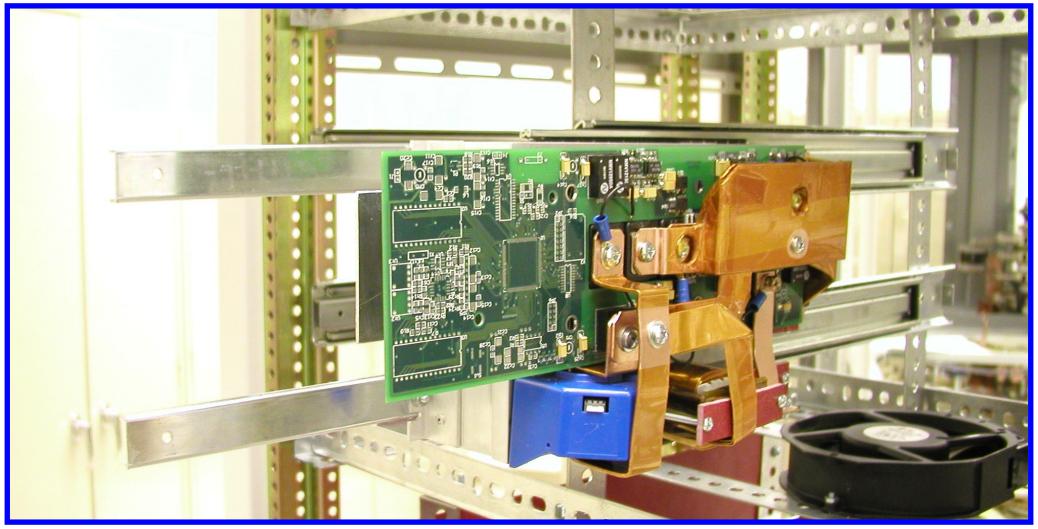
Plug & Play Capability

- PEBB approach enhanced by providing PnP capability
  - Mounted on rails for easy replacement & reconfigurations
  - Full compatibility between soft-switching and IPM-Based PEBB's
  - Unique slot ID for easy reconfigurations
  - Hot-swappable control and communications
- Software PnP

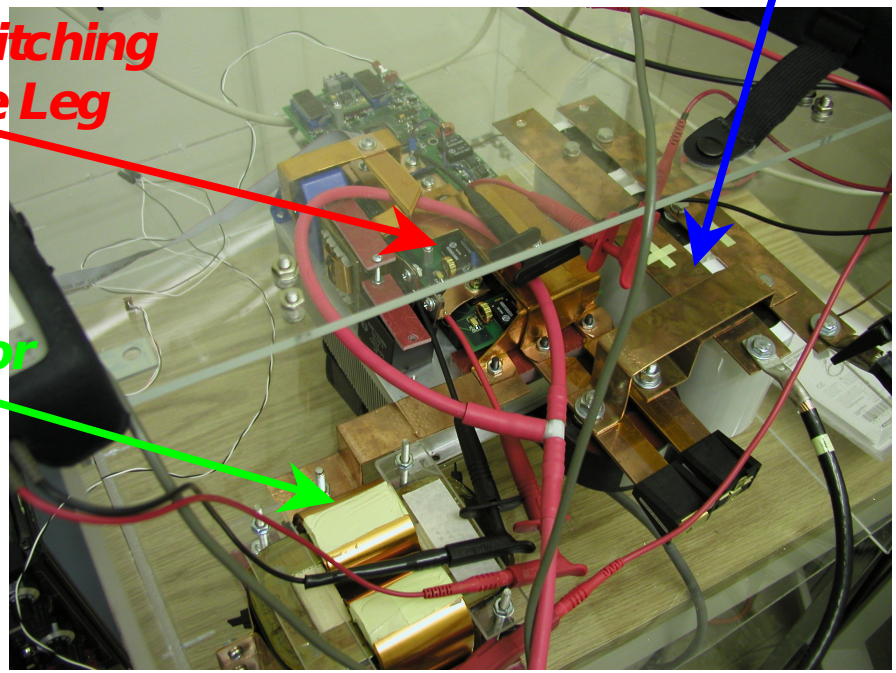


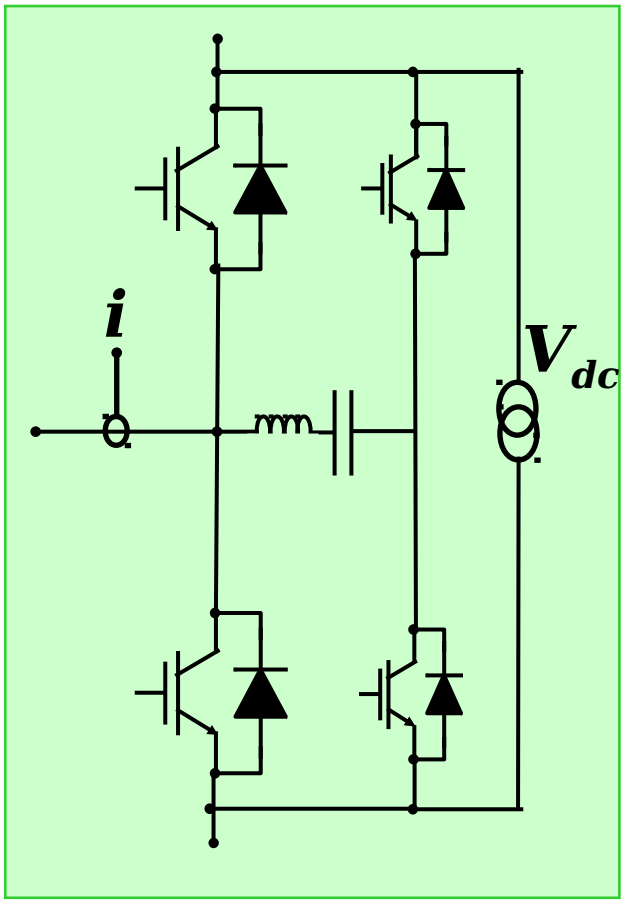
Hardware Under Development

- 100 kVA/800 Vdc demonstration testbed
  - Four 35kVA soft-switching PEBB phase-legs completed
  - Four 35kVA IPM-Based PEBB phase-legs being built
- Reconfigurable topologies
- Plug and Play capability



Soft-Switching Phase-Leg PEBB

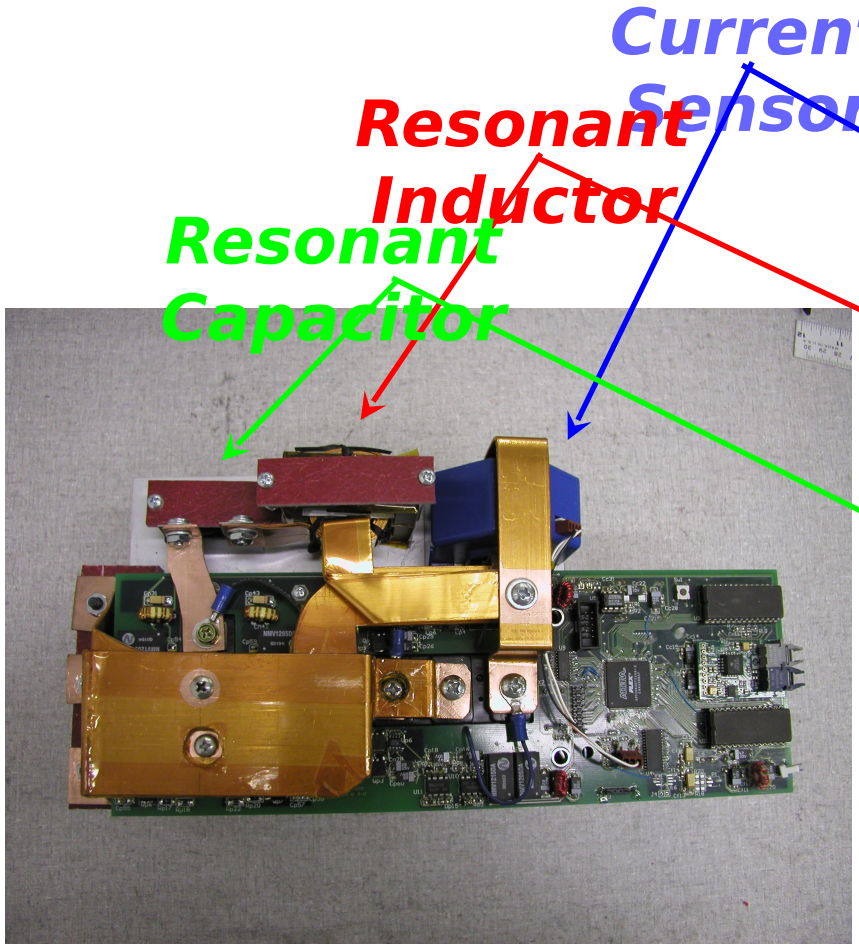


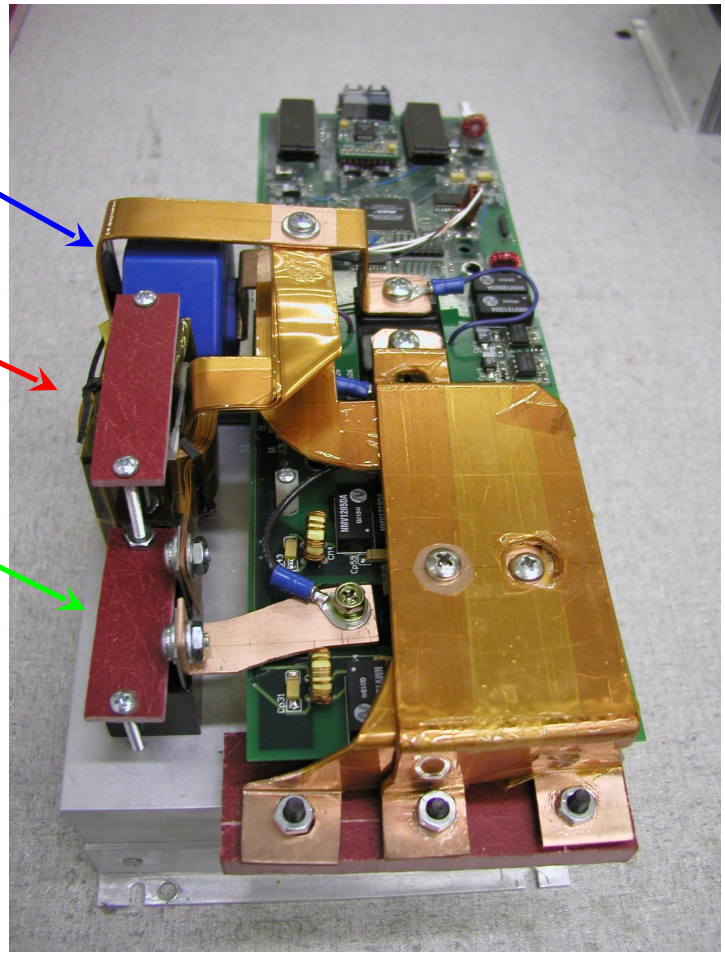


- Four phase-legs have been built and tested
  - Improved ZCT (IZCT)
  - ZV-ZCT

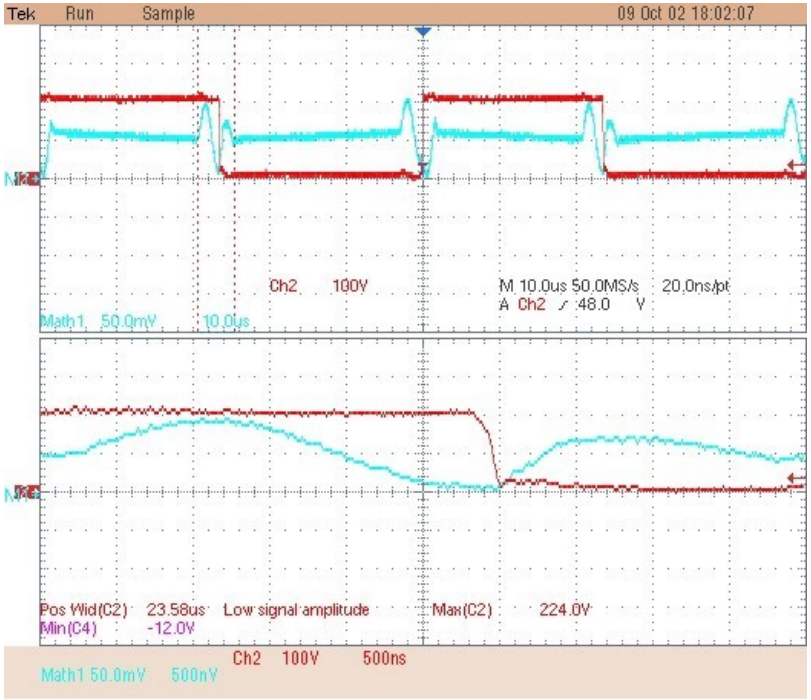
Soft-Switching PEBB

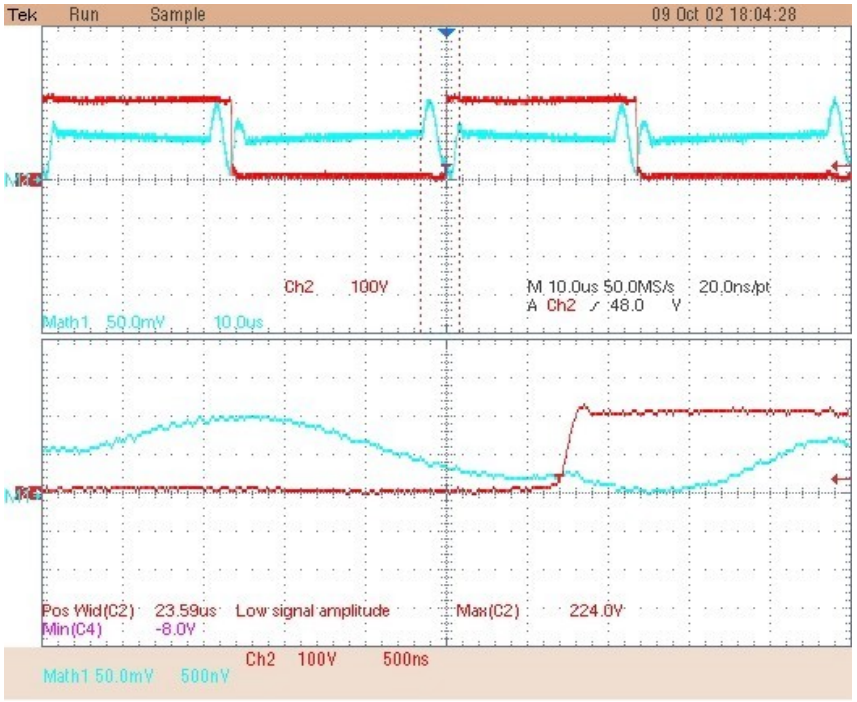
Soft-Switching Phase-Leg PEBB





Test Results: I-ZCT Waveforms



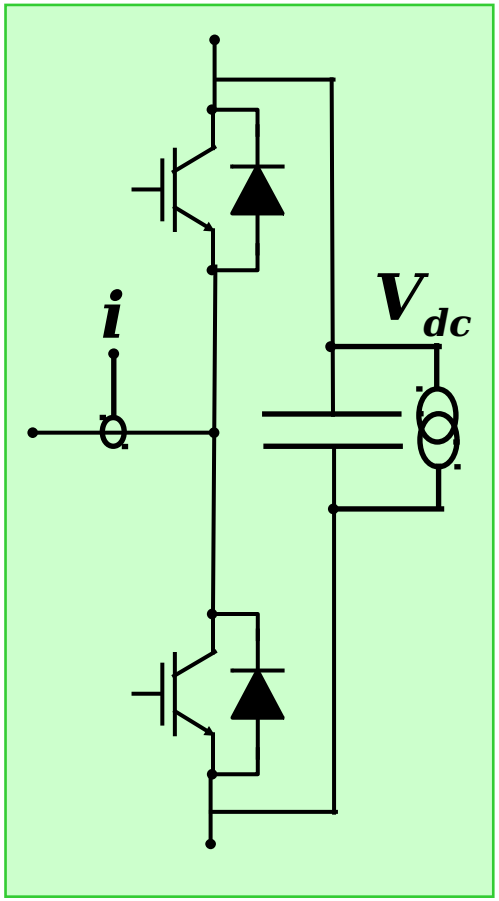


Turn-on Transient

Turn-off Transient

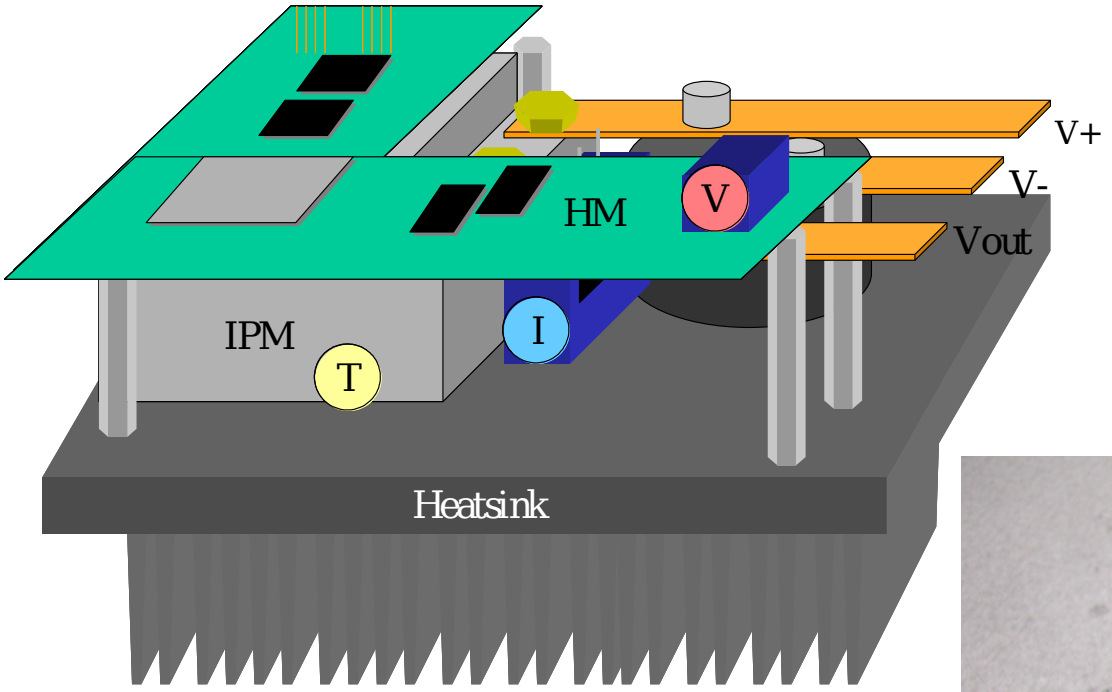
IPM-Based Phase-Leg PEBB

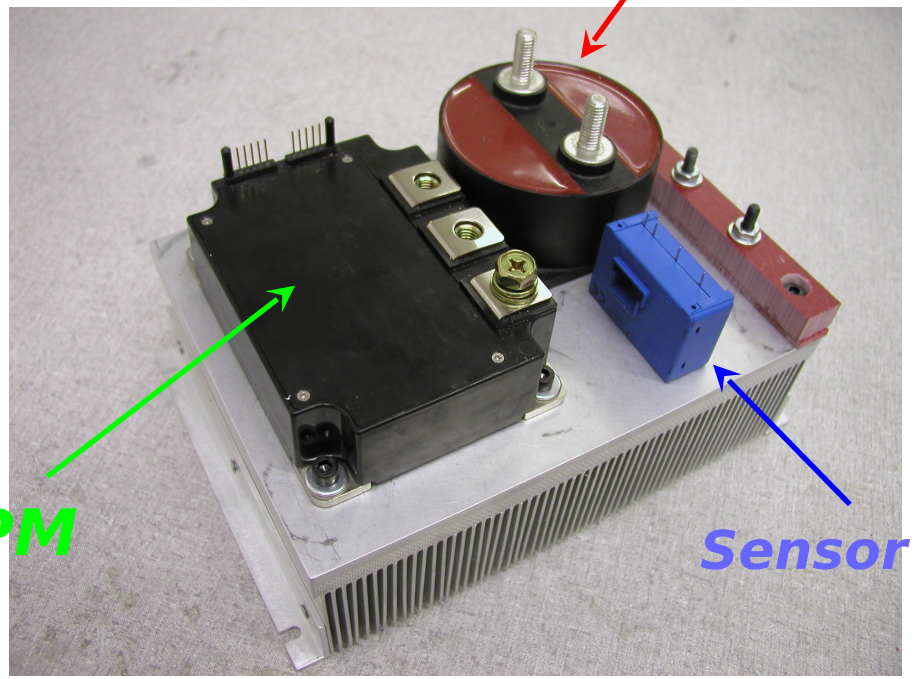
- Employs IPM technology
- Integrated dc-link capacitor in phase leg
- Integrated voltage and current sensors
- Hard-switching
- Built-in protection



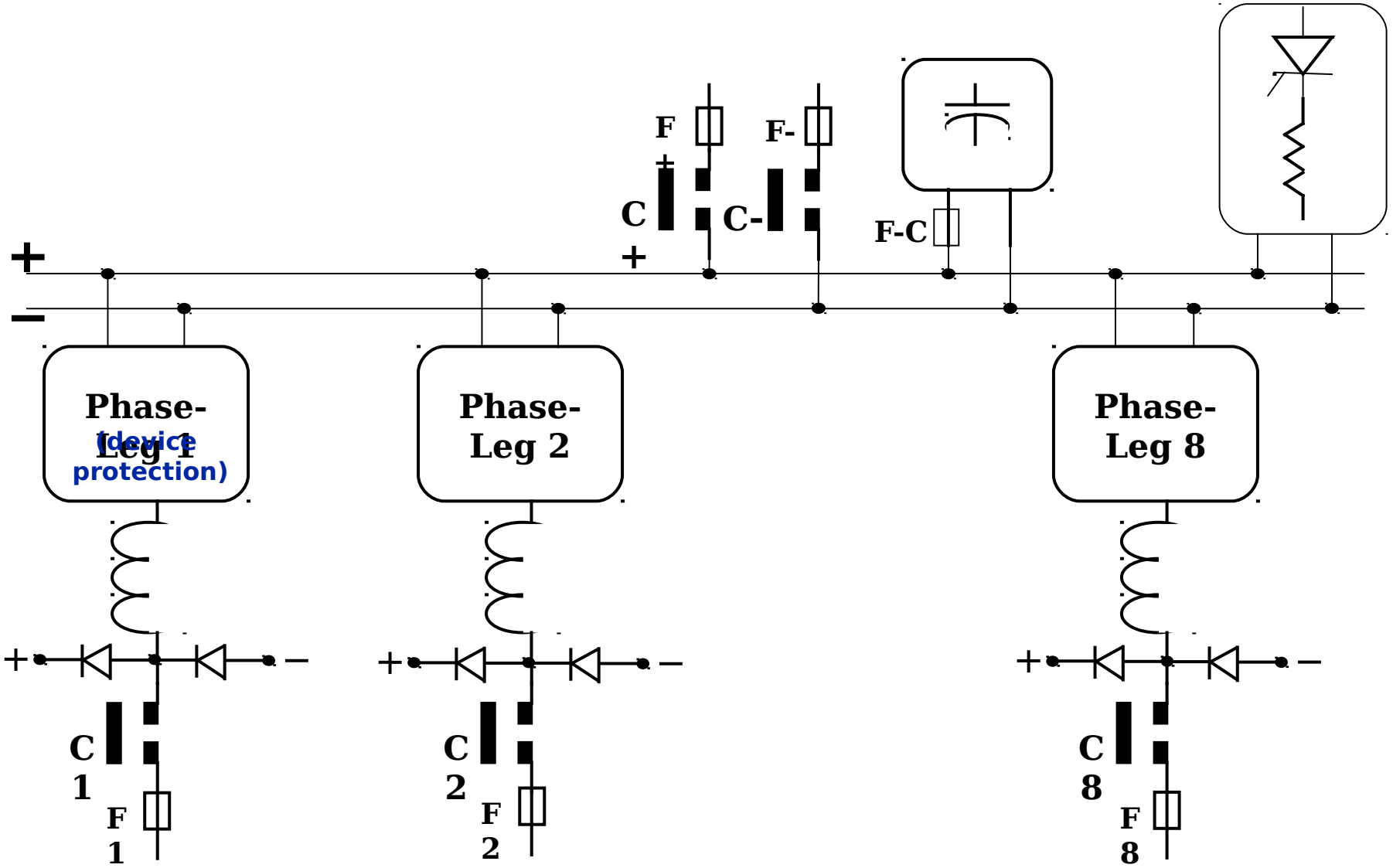
IPM-Based PEBB

IPM-Based Phase-Leg





Protection Scheme



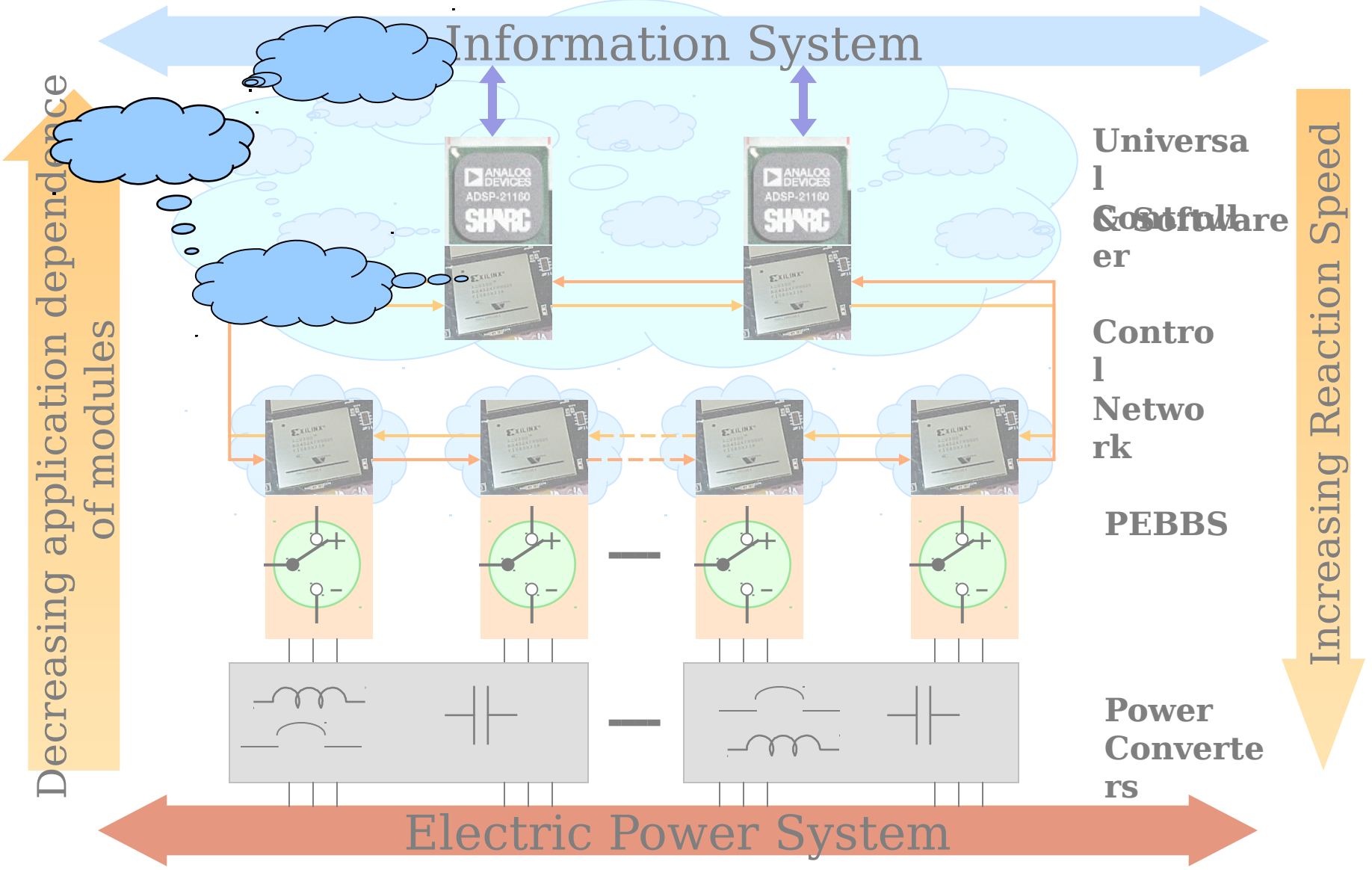
System Demonstration

- Full system built by end of year
- Start 3 phase, 100 kVA system testing next year
- Demonstrate full function, plug and play features and capabilities

Presentation Outline

- **System Architecture**
  - What is PnP
  - What does the PnP architecture look like?
- Control Hardware
  - Digital hardware to support control functions
- Power Hardware
  - Demonstration testbed
- **Control Software** — Stephen Edwards
  - PnP modular software architecture

PnP System Architecture



Software Goals

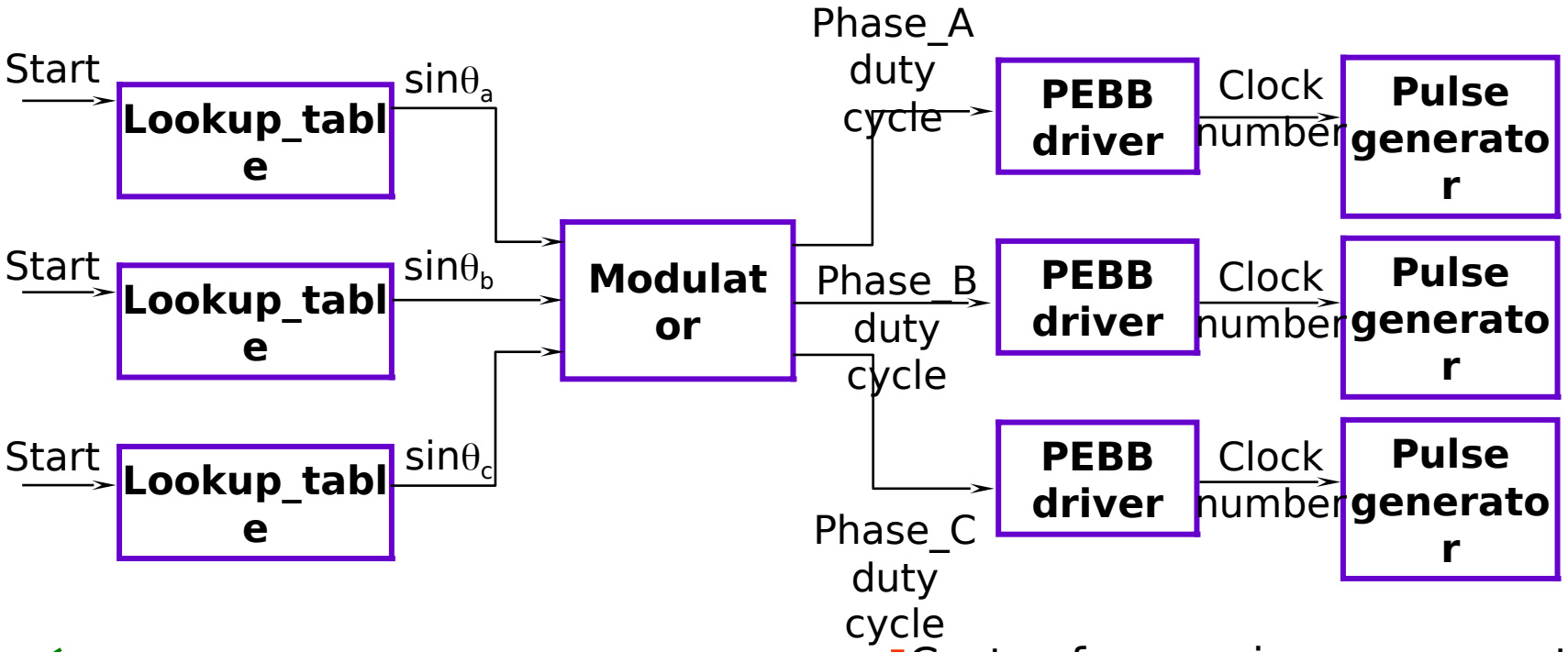
- Achieve same Plug and Play goals when constructing control software:
  - Use software building blocks that are:

    modular, reusable, reconfigurable, flexible,

    application-independent
  - Minimal coupling, encapsulated hardware dependencies, support for transparent distribution
- Dataflow architecture
- Library of Elementary Control Objects (ECOs)

Sample PEBB Control App:

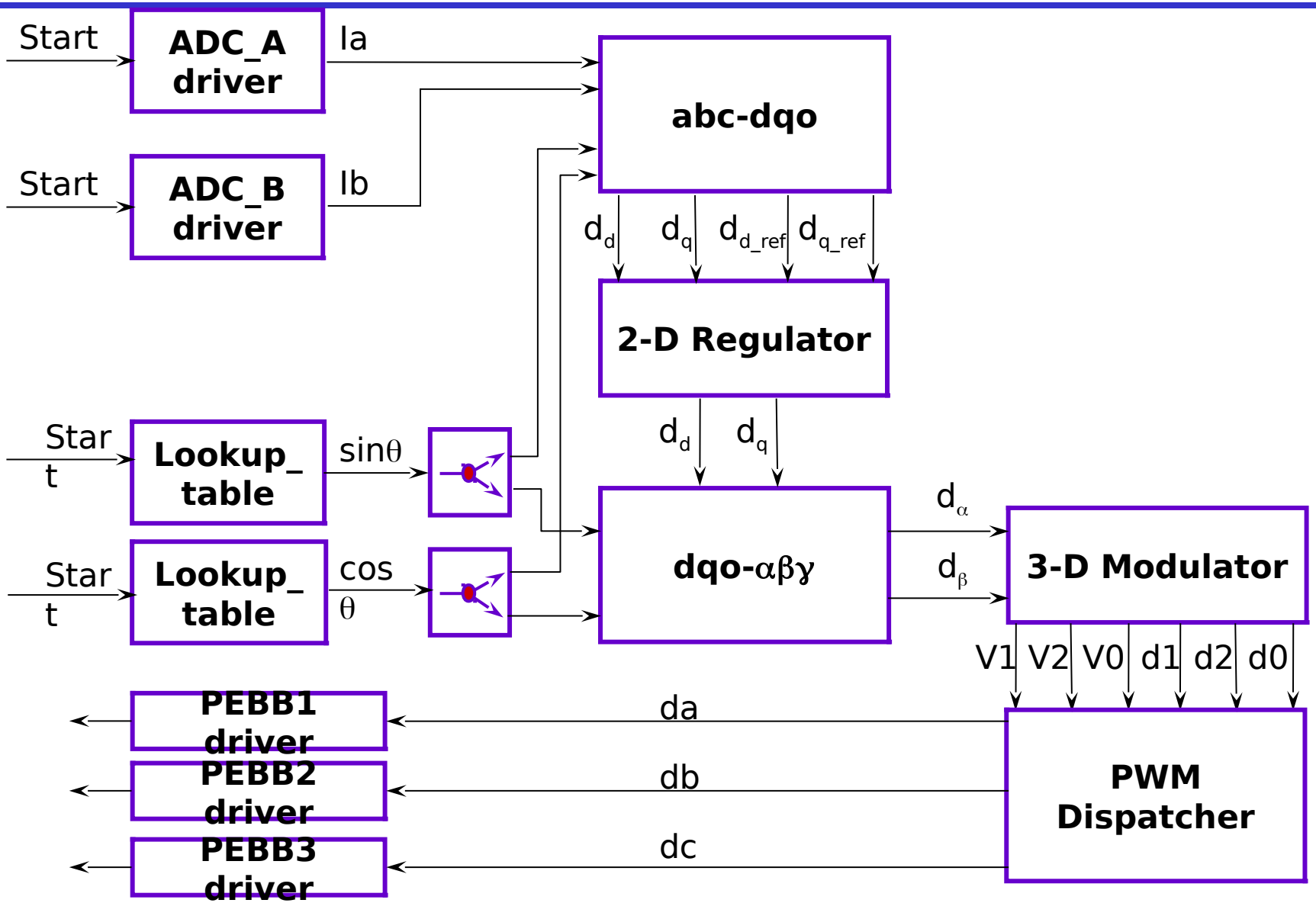
Open loop 3-phase inverter



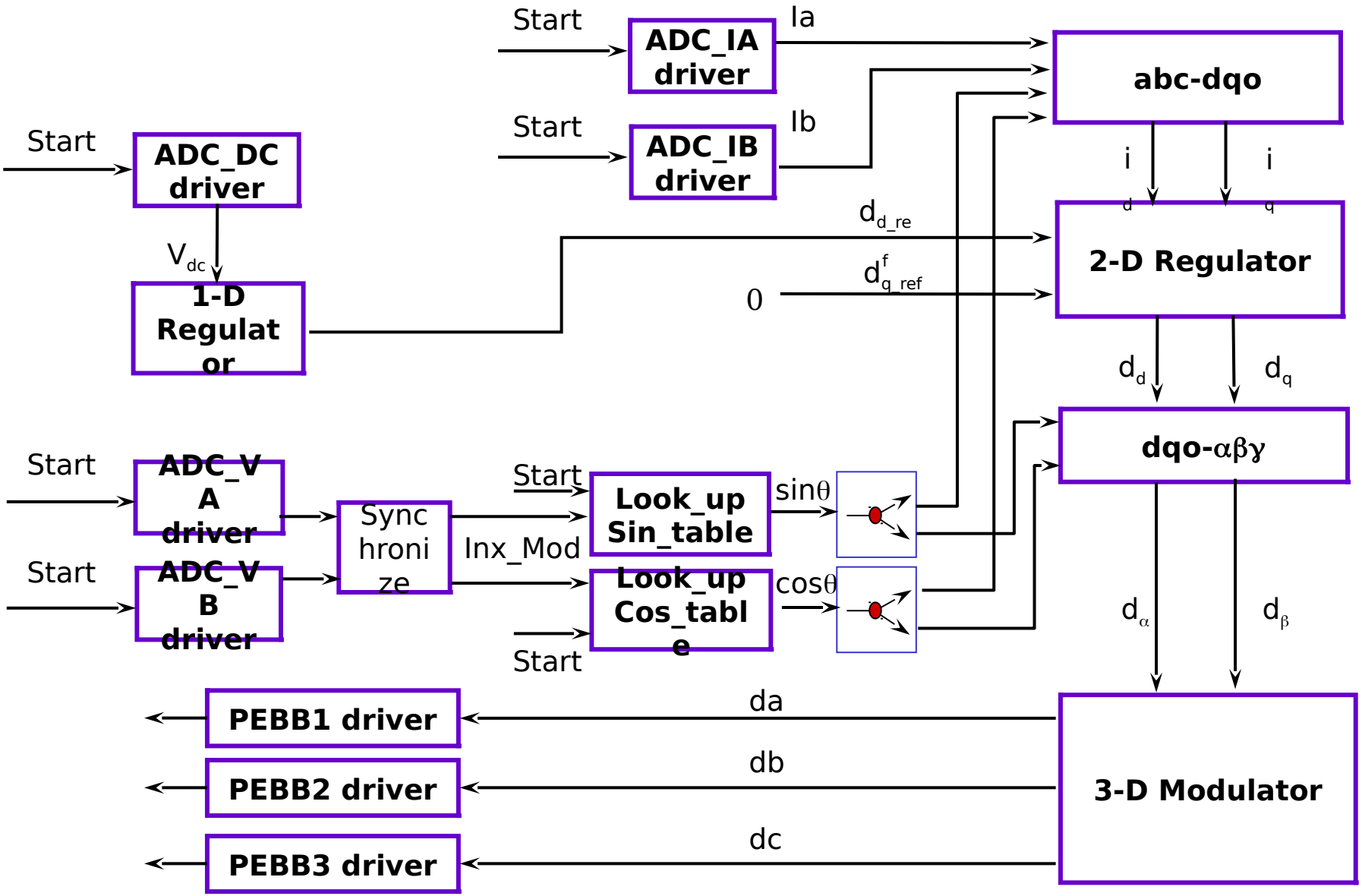
- ✓ Reusability
- ✓ Reconfigurability
- ✓ Natural mapping onto distributed software execution

- Costs of managing concurrent dataflow processes
- Costs of managing data channels
- Granularity of dataflow processes

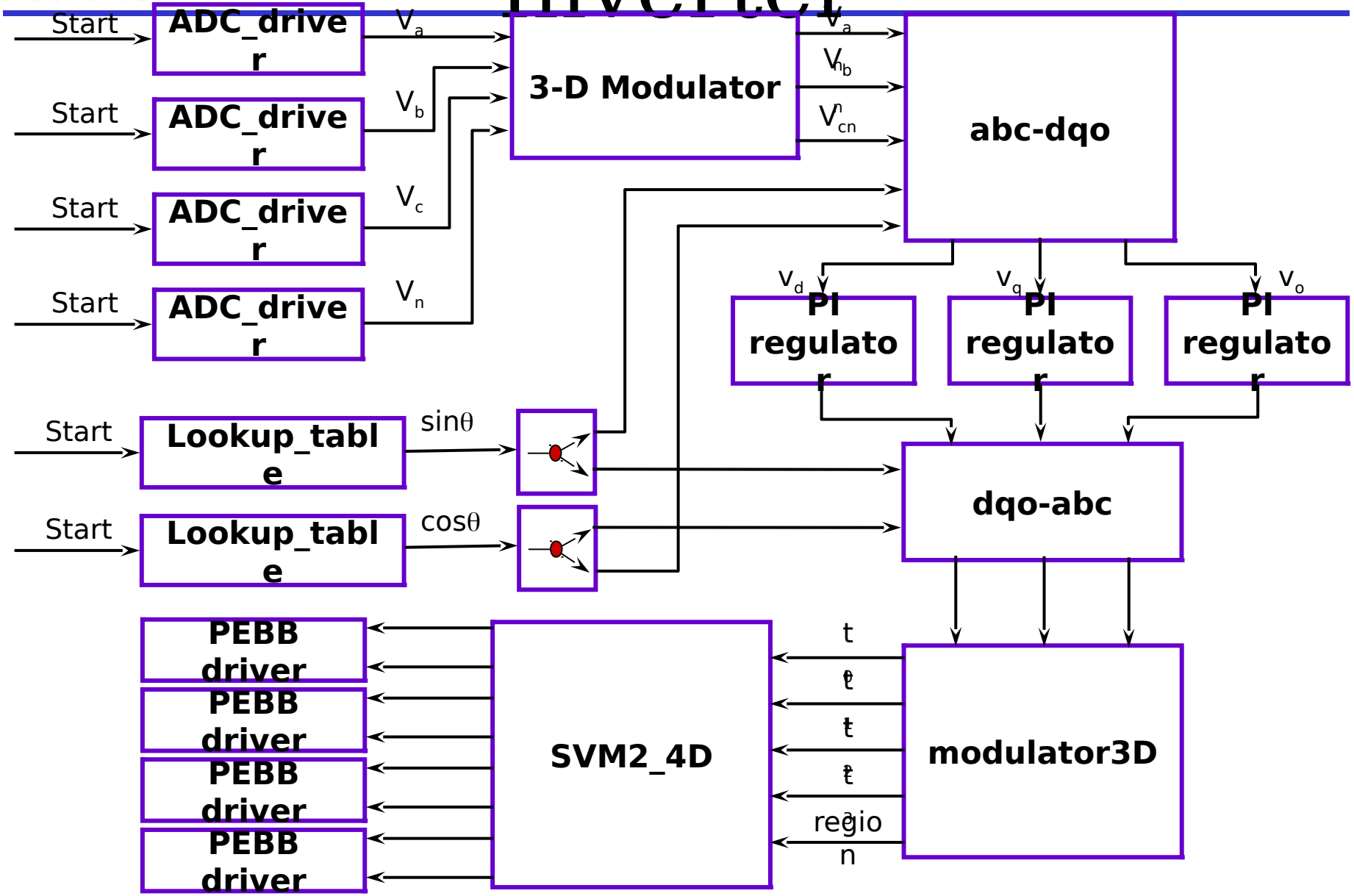
Closed Loop 3-phase Inverter



3-phase PEBB-based Rectifier



4-leg PEBB-Based Inverter



Dataflow Architecture Real-time Kernel

Need for DARK:

- Dataflow applications
  - Contain many small processes and lots of inter-process communication
  - Require special scheduling
- Traditional RTOSes impose too much overhead

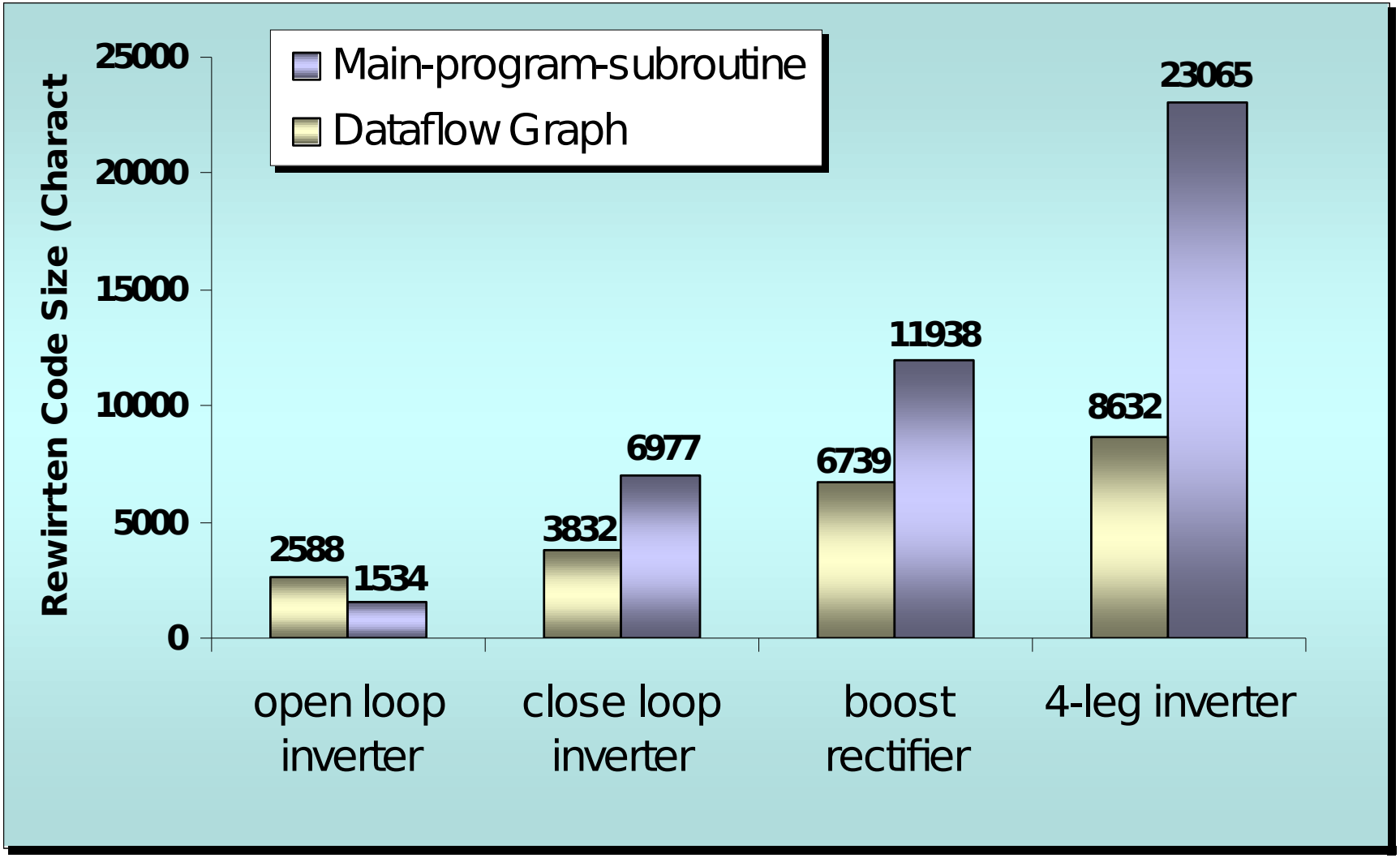
Features of DARK:

- Faster and fewer context switches
- Efficient inter-component communication
- Scheduling for dataflow based on incoming data
- Dynamic priorities
- Real-time support

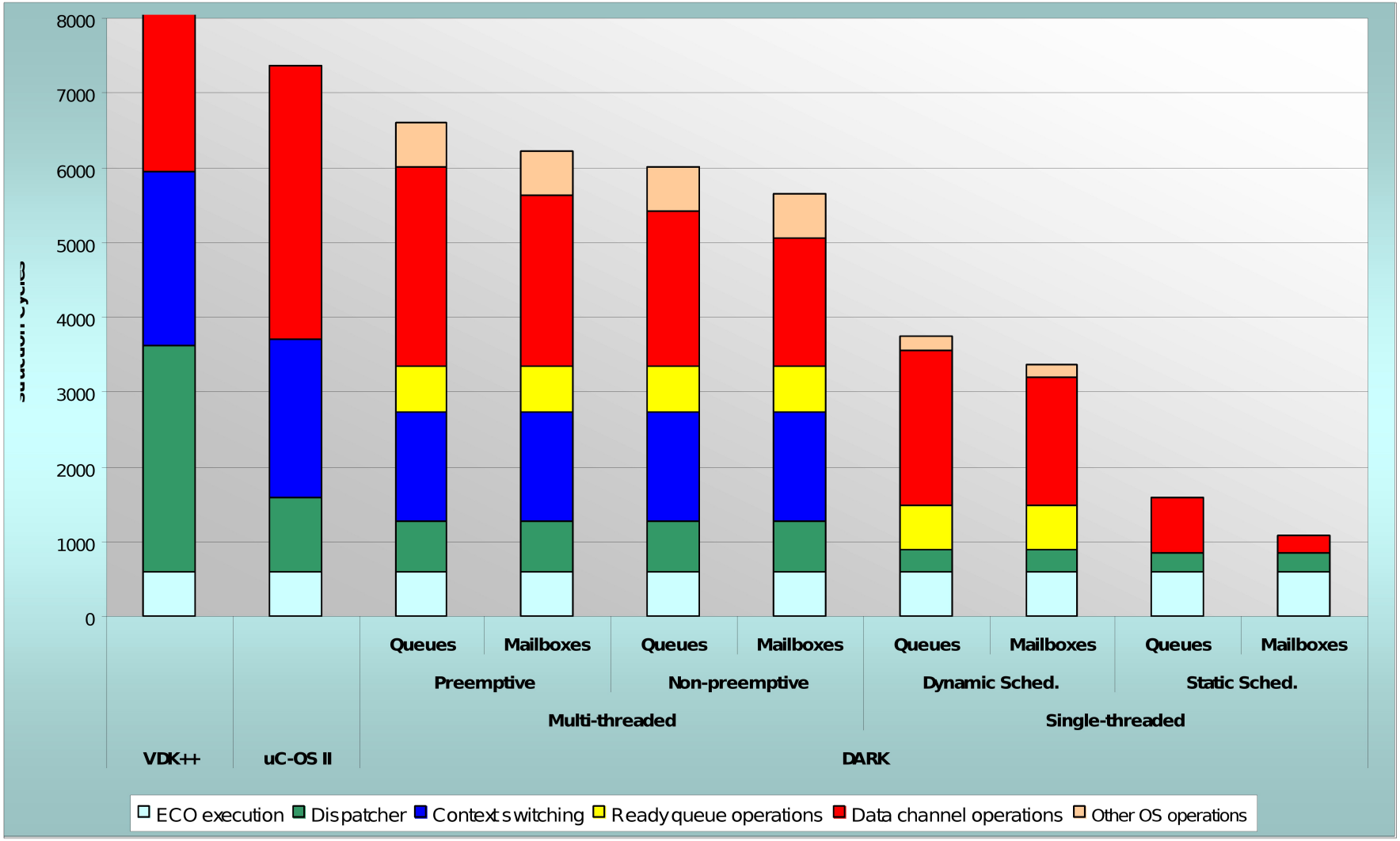
Under Development:

- Object-oriented C++ version
- Transparent distribution support

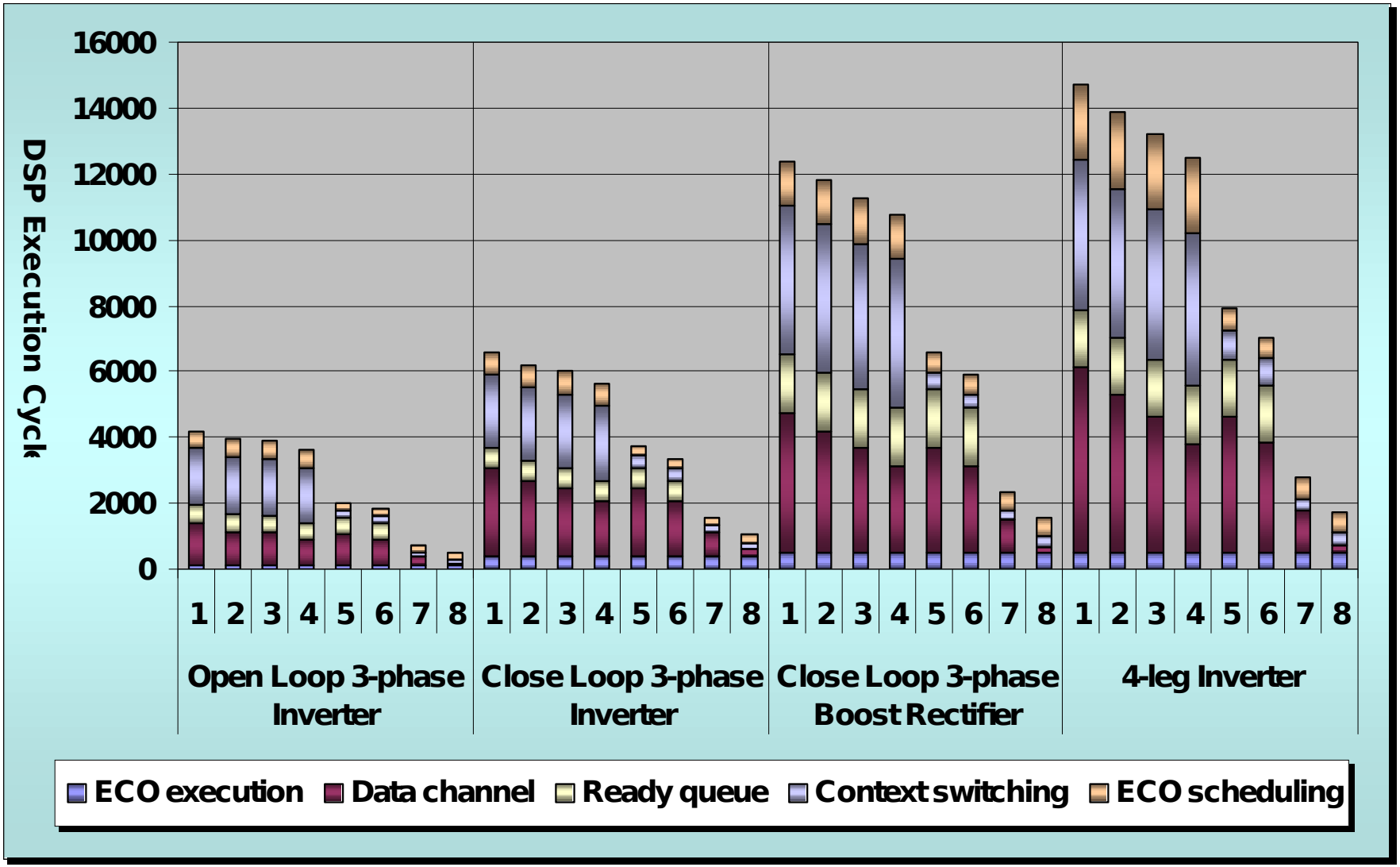
Code Size Comparisons



DARK Performance



Performance Across Apps



S/W Architecture Results

Advantages:

- Provide a framework to support modularity, reusability and reconfigurabi
- Reduce overall development cost and time
- Decrease the complexity of development and testing
- Minimize redesign effort
- Allow designers to focus easily on key design areas
- Make system expansion and upgrade easier

Disadvantages:

- Need more run time resource
- Require more runtime support
- Restricts optimization of ECOs based on application needs

Conclusions

- Pursued PEBB PnP architecture principles in both hardware and software
- PnP application manager (univ. controller) and hardware manager modules
- Demonstration test bed with 8 PEBBs (interchangeable hard- and soft-switching), modular interconnect and ID system
- Modular flexibility, reconfigurability, and reuse for power stage PEBBs, control hardware, and control software